



ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

**Electronic Notes in
Theoretical Computer
Science**

Electronic Notes in Theoretical Computer Science 258 (2009) 35–49

www.elsevier.com/locate/entcs

Quantum Fault Tolerance in Systems with Restricted Control

Joseph Fitzsimons¹*Department of Materials
University of Oxford
Oxford, United Kingdom*

Jason Twamley

*Centre for Quantum Computer Technology
Macquarie University
Sydney, NSW 2109, Australia*

Abstract

In many proposed architectures for quantum computing the physics of the system prevent qubits from being individually controlled. In such systems universal computation may be possible via bulk manipulation of the system. Here, we describe a method to execute globally controlled quantum information processing which admits a fault tolerant quantum error correction scheme. Our scheme nominally uses three species of addressable two-level systems which are arranged in a one dimensional array in a specific periodic arrangement. We show that the scheme possesses a fault tolerant error threshold.

Keywords: quantum computation, fault-tolerance, global control.

1 Introduction

Global control provides a novel way of quantum computation which should greatly reduce the complexity of classical control technology required in a medium-to-large scale quantum processor. A globally controlled quantum computer architecture typically only permits one to apply quantum gates homogeneously on large subsets of the processor and one is not allowed to target gates on individual qubits within the processor. A number of designs have appeared in the literature but so far their usefulness has been hampered by the lack of any design which also incorporates globally controlled quantum error correction executed in a fault tolerant manner. In this note we describe a 1D scheme for such fault tolerant computation which only

¹ Email: joe.fitzsimons@materials.ox.ac.uk

includes three addressable qubit species arranged in a self-similar one dimensional pattern.

1.1 Background

The study of globally controlled architectures began with [20], which used a three species spin chain arranged in a periodic linear array. In [2], a two species 1D design was developed where an “always-on” interaction was modulated by homogenous local unitaries (HLUs). Models where one can homogeneously modulate the inter-chain couplings were presented in [3,19]. Both [7] and [13], examine the simulation power of a quantum system with always-on interactions and modulated HLUs. More recently, a number of globally controlled schemes for 1D quantum computation have been discovered displaying various levels of sophistication of construction and control [14,12,21,27,8].

Most of these globally controlled schemes can be cast into two main categories, (A) those that use special “software labels” (the *control unit*, in [2]), which move via global pulses within the processor and whose purpose is to effectively localise an applied global pulse to a local region [2,4,27]. The other main category (B), is where one uses “hardware labels” to trigger the conversion of globally applied pulses to qubits within the device. In [22] this is achieved via a change in global parity of an evolving delocalised qubit pattern upon impacting with the physical ends of the chain, while in [8], control is achieved by manipulating the delocalised qubit pattern when it also impacts an end of the chain. One faces a number of challenges in developing a fault tolerant quantum error correct scheme in a 1D globally controlled design with nearest-neighbour (n-n) interactions.

From [1,10,17], to implement concatenation of QEC one at least requires (1) fully parallel execution of quantum computation, (2) one must ensure that errors do not proliferate, i.e. one round of computation and error correction is successful in reducing the overall error rate, (3) methods to remove entropy from the system, and (4) the error rates do not drastically increase with the concatenation level. Obviously the restriction to n-n models and the associated increased error rates due to the shuttling of qubits around to execute long range gates will prove detrimental to the performance of fault tolerant quantum error correction but a number of works have now shown that FTQEC is still possible [1,10,9,24,26,25]. We will restrict ourselves below to category (B) designs where one has a hardware trigger to manipulate the delocalised qubits. Previously a number of works have examined possible FTQEC schemes for category (A) designs [6,15,16], but there one has difficulty in correcting for errors in the special “software label” itself using only global control.

1.2 Outline

We follow [8], where one uses an always-on Ising (ZZ), interaction and a single qubit Hadamard HLU to construct a global operation $\mathcal{S} = \overline{H} \cdot \overline{CZ}$, as a mirror iterate and via edge operations, one fashions universal quantum computation. Here \overline{H} denotes a Hadamard operation applied to every qubit in the chain, while \overline{CZ}

denotes controlled phase operations applied uniformly between neighbours. Below we shall assume that the spin chain is subject to temporal errors (which we assume to be independent Pauli errors), and we construct a hierarchy of logical meta-qubit encodings, such that on the highest level we effectively will have a meta-Ising model where the associated meta- ZZ , meta- \bar{H} and meta- \bar{CZ} , operations possess greatly reduced error rates. We then can implement quantum computation on this meta-cellular automaton using either of the approaches of [22,8]. We will argue that the resulting model possesses a fault tolerant threshold, however we will here not give specific estimates for the magnitude of the threshold, limiting ourselves to providing a proof of existence.

2 Requirements

In order to perform quantum error correction, a supply of cold ancillae is required. If the supply of available qubits is limited, then we must reuse ancillae, by resetting them to the $|0\rangle$ state. Now, consider a uniform spin- $\frac{1}{2}$ network. If the chain is comprised of only one species then all qubits must be reset together, destroying the quantum computation currently in progress.

To achieve error correction, it is therefore necessary to have at least four distinct energy levels for storing quantum information and a mechanism for resetting two of these levels to a specific state. This reset ability will most easily be achieved by accessing an additional energy level which spontaneously decays within a short time interval. The storage levels correspond to the distinguishable ancillary and logical qubits. This can be achieved either using qudits at each site with four or more levels, or by using more than one species of qubit. In the discussion that follows, we will take the latter approach, considering a chain of qubits with a regular pattern of A and B species qubits. We will implement the reset operation using a third species C which possesses a spin non-conserving transition which can be triggered via an appropriate global pulse sequence to reset the C -species to the $|0\rangle_C$ state irrespective of its initial state. While it is likely possible to implement fault tolerant computation using only two species, using a third species will greatly simplify our design.

In [8], we considered a setup where the ends of the chain were assumed to be addressable separately from the bulk of the chain due to having only a single nearest neighbour. Unfortunately it is not appropriate to use end systems as an entropy sink since the rate at which errors occur in the chain is a function of the chain length, while having fixed the entropy sinks to be at the end of the chain, it is only possible to remove entropy from the system at a constant rate. In what follows we will consider A , B and C to be separate physical species.

3 Propagation of errors

The first step in understanding how error correction can be performed within the chain is to understand how errors propagate within subchains. It has already been

shown in [8] that it is possible to implement quantum computation in a uniform spin chain, by applying successive rounds of the iterate \mathcal{S} and performing occasional local operations on the end of the chain. We will consider the same system, extended only by the addition of two additional species. We will use the convention that the species an operation is to be performed on will be denoted by a subscript. In such a system, it is useful to define $|0\rangle_L^{\otimes N}$ as $|0, +, 0, +, \dots\rangle$.

We will consider all errors which occur to be random Pauli operators. We are justified in taking this approach as any operation on the system can be written as a sum of Pauli operators, weighted by complex coefficients, and the identity. Additionally, the application of an error detection procedure generally projects the encoded qubit onto either an uncorrupted state or onto one which has a Pauli group operator error present. Here, for simplicity, we will focus on the effect of single qubit errors, although more general errors can be treated in the same way.

The operations performed on the spin chain by successive applications of the iterate are Clifford group operations, and so it is possible to track the evolution of quantum states and operators efficiently by the Gottesman-Knill theorem[11]. We will use the Heisenberg picture to track the effect of multiple applications of the iterate on the error operators. These Pauli operators evolve according to the rules

- (1) $\overline{CZ}\sigma_z^{(a)} = \sigma_z^{(a)}\overline{CZ}$
- (2) $\overline{CZ}\sigma_x^{(1)} = \sigma_x^{(1)}\sigma_z^{(2)}\overline{CZ}$
- (3) $\overline{CZ}\sigma_x^{(N)} = \sigma_z^{(N-1)}\sigma_x^{(N)}\overline{CZ}$
- (4) $\overline{CZ}\sigma_x^{(a)} = \sigma_z^{(a-1)}\sigma_x^{(a)}\sigma_z^{(a+1)}\overline{CZ}$
- (5) $\overline{H}\sigma_z^{(a)} = \sigma_x^{(a)}\overline{H}$

and so the propagation of errors within the chain follows a similar delocalisation as logical qubits. We should note, however, that errors occurring midway through a mirror cycle become delocalised at the time when the logical qubits revive after the mirroring. This situation is illustrated in figure 1.

The key to detecting and correcting errors rests on the fact that individual errors introduce a change in the state of the end spins of the chain only once per mirror cycle. By using standard error correction codes to detect these changes in the state of the end spins over two mirror cycles it is possible to locate the error, as we will see.

4 Base concatenation level

We first turn our attention to how one level of error correction can be performed, before moving on to discuss how this scheme can be concatenated to arbitrarily high levels. As we will see, many of the operations are performed in the same manner at both the base level, and at higher levels. There are, however, a number of important differences.

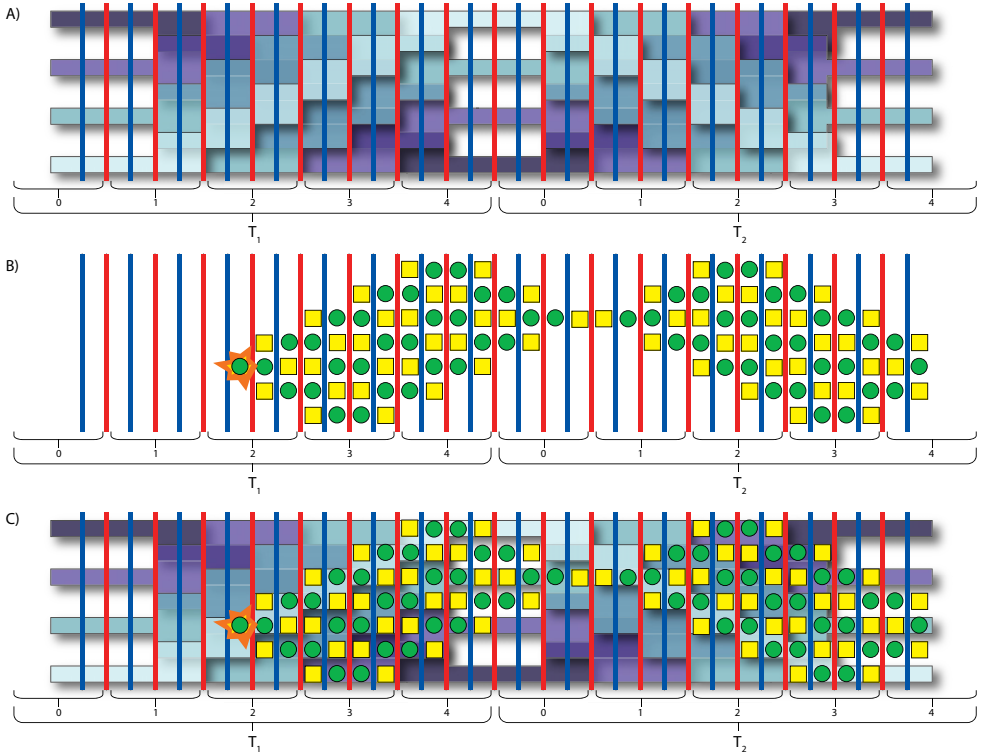


Fig. 1. Propagation of errors. A) This plot shows how the logical qubits, shown in various shades of blue, become delocalised over the chain while undergoing two complete mirror cycles. The buffer qubits are taken to be in the $|+\rangle$ state. B) Here we see a random error occurring and being propagated through the spin chain. C) This is a combination of the two previous plots showing the relative localisation of the Pauli operator introduced by the error and the logical qubits.

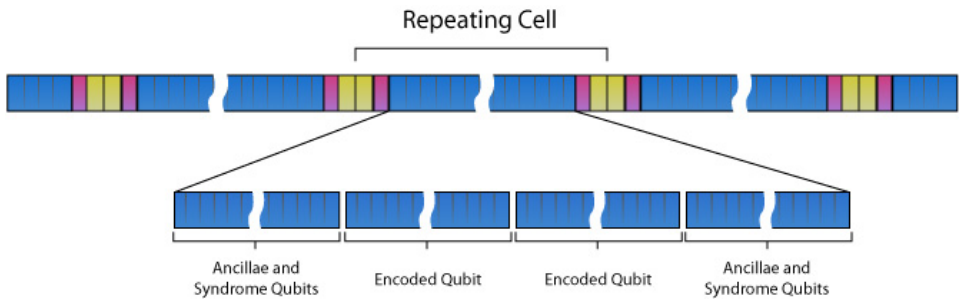


Fig. 2. Illustration of the base level of encoding. Species *A* (blue) spins are used to encode two qubits and their associated ancillae back to back, respecting the mirror symmetry of the system. Species *C* spins (pink) are used for reset and local control. Together with species *B* (yellow), these form a chain connecting encoded qubits.

4.1 Chain layout

For the first level of error correction the chain layout is quite simple, consisting of long chains of species *A*, separated by a sequence of four spins of species *B* and *C*. The required layout is depicted in figure 2. The species *A* subchains are used to encode two logical qubits back to back. The ancillae required for non-Clifford

operations and for syndrome extraction are included on either side of the encoded qubits. As we will see, it is possible to perform any mirror symmetric computation within these subchains, while asymmetric operations are not possible. For this reason, the encoding of the two qubits needs to be mirror symmetric about the centre of the chain. The B - C segments are used to connect pairs of encoded qubits to allow for entangling gates. Species C spins will be used to reset the ancillae, when required.

4.2 Computation within species A and B subchains

From [8], given control over the end spins of a uniform Ising spin chain it is possible to perform universal quantum computation. If decoupling pulses are applied to the species C spins, the species A and B subchains become essentially independent Ising spin chains. It is also possible to decouple a subchain by stroboscopically applying X gates to the entire chain. These will commute with the $\sigma_z^{(i)} \sigma_z^{(i+1)}$ term of the Hamiltonian, providing that both spins i and $i + 1$ lie within the subchain. The coupling between the end spin and the first spin of the next species, however, will be averaged out to zero, since the X gate anticommutes with the coupling term in the Hamiltonian. This allows the decoupling of species A and C and the coupling between B and C to be controlled separately.

In order to achieve universal computation within these chains, it is necessary to have local control over the ends of the chains. To achieve this, we set the species C spins to the $|0\rangle$ state, using the associated global reset pulse. In order to perform local rotations on the end of the chains we temporarily halt the decoupling of the interaction, between the target subchain and species C , for some time τ . The Ising interaction leads to an operator

$$(6) \quad U_{rot}(\tau) = e^{-iJ\sigma_z^{(C)}\sigma_z^{(1)}\tau}$$

being applied to the chain. As species C is in an eigenstate of σ_z , this simplifies to

$$(7) \quad U_{rot}(\tau) = e^{-i(J\tau)\sigma_z^{(1)}},$$

corresponding to a local rotation on the end of the subchain, as desired. It should be noted that due to the symmetry of the chain, the same rotation must be applied to both ends of the subchain, and identically to all subchains of the same species. This mirror symmetry requirement restricts the available operations possible on the chain to those which are mirror symmetric. Thus, while asymmetric operations are impossible, we have a prescription for performing all mirror symmetric operations.

In order to implement fault tolerant computation two main building blocks are required. The first is a method for detecting and correcting or otherwise neutralising errors which occur during the computation. The second is a way of performing quantum gates on the encoded qubits which do not propagate errors within an encoded qubit. We will now turn our attention to the first of these requirements, and discuss how errors can be detected and corrected within our chain.

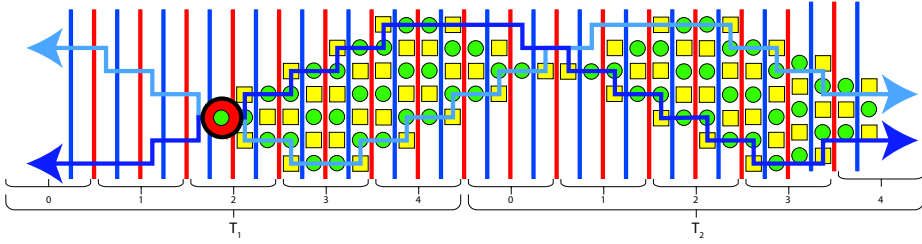


Fig. 3. Error detection. This is the error propagation pattern from figure 1. By propagating backwards (or forwards) from the detection points it is possible to determine where the error can be corrected with a single Pauli operator.

4.3 Error detection

As described in [8], logical qubits will only interact with other qubits when they have delocalised over the edge of the chain. The situation is similar for errors, since they have identical propagation patterns. If we apply conventional error detection techniques within our chain over the course of two mirror cycles, we will detect errors at two separate time slots, once in each cycle. It is necessary to record both of these, and so double the number of syndrome bits are produced. The exact form of the error detection sequence will of course vary depending on which encoding is chosen, but will always be possible since it falls into the class of mirror symmetric operations due to the back-to-back layout of the species *A* subchain. It is possible to detect the spatiotemporal location of where the error has occurred by tracing the path of a single Pauli operator corresponding to the measured error backwards from each point of detection. This situation is shown in figure 3. The error is then located at the point where these paths overlap. Due to the mirroring action of the iterate, these error locations will be periodic in time. Applying a correcting single qubit Pauli operator at any of these locations will cancel the error.

4.4 The error correction cycle

As with error detection, the exact form of the gate sequence depends largely on the specific error correction code we choose to implement. As the qubits have been encoded in a mirror symmetric manner, however, the error correction cycle must also be mirror symmetric. Since all symmetric operations are possible within the chain, any error correction procedure must also be possible. In general, we expect the error correction procedure to begin with syndrome extraction, as in the previous section. Following this correction operators are applied conditioned on the syndrome qubits. As at least two syndrome qubits are necessary to locate the error, this will in general require performing multiply-controlled Pauli gates on the subchain at each time step. Once the correction has been accomplished, the syndrome qubits are swapped, one at a time, onto the species *C* spins, where they are reset to $|0\rangle$.

4.5 Fault tolerant gates

Our second component of a fault tolerant architecture is a mechanism to perform quantum logic gates upon encoded qubits without propagating errors within these

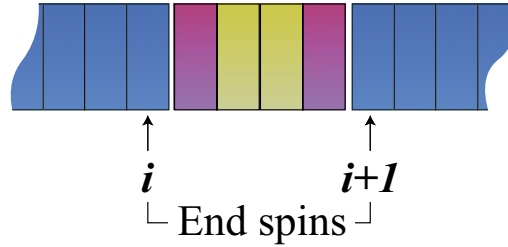


Fig. 4. Illustration of the location of end spins of subchains utilised in entangling gates.

encoded qubits. While the exact details of these gates depend on the specific error correction code in use, we will discuss how the basic building blocks of such gates can be built up, and mention specifically how these can be achieved when using the seven qubit Steane code[23].

4.5.1 Hadamard and other local Clifford gates

While not all codes are the same, the class of stabiliser codes does have some common properties. In any stabiliser encoding, Pauli operators on the logical qubits can be implemented through some combination of local Pauli gates on each of the underlying qubits. It is possible to implement each of these local gates in the same manner as local gates were achieved in [8].

Many stabiliser codes also allow for Clifford group operations to be applied in the same kind of transversal manner. The Steane code is particularly helpful in this context, as it is possible to apply a Hadamard gate to the logical qubit by applying Hadamard gates transversally to the underlying qubits. Fortunately, and somewhat surprisingly, this corresponds to a single global pulse.

4.5.2 Non-Clifford group gates

In most quantum error correction schemes, it is not possible to perform non-Clifford group gates transversally, and so ancillae are used to prepare an entangled quantum state, which is then used to implement non-Clifford group gates. A notable exception to this rule, however, is the 15-qubit quantum Reed-Muller code, which does allow for transversal non-Clifford group gates[18,5]. If ancillae are required, then they can be prepared as usual, since universal computation is possible within each subchain.

4.5.3 Controlled-Z gates

In order to perform full computation an entangling gate such as a CNOT or controlled-Z gate is required. As we have discovered, all mirror symmetric local operations within the subchain are possible. This allows us to perform pairwise entangling operations between the encoded qubits within the same subchain. As all local operations within the subchain are controlled by z -axis rotations on the end spins, a CZ operation between the ends of adjacent species A subchains will suffice to perform any entangling gate.

To perform a CZ between neighbouring chains we first reset the B - and C -spins to the $|0\rangle$ state. Next a CNOT is performed between the end spins of the species A chain (see figure 4) and the species C spins. The state of the B - C chain is now dependant on the the states of the end of the chain as follows:

State of end spins	State of chain
$ 0_i\rangle 0_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$ 0_i\rangle 1_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 1\rangle$
$ 1_i\rangle 0_{i+1}\rangle$	$ 1\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$ 1_i\rangle 1_{i+1}\rangle$	$ 1\rangle \otimes 0\rangle \otimes 0\rangle \otimes 1\rangle$

The next step is to mirror invert the B - C chain. The easiest way to accomplish this is to apply $(H_B H_C S)^5$ with species A decoupled. The subsequent state of the chain is given below.

State of end spins	State of chain
$ 0_i\rangle 0_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$ 0_i\rangle 1_{i+1}\rangle$	$ 1\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$ 1_i\rangle 0_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 1\rangle$
$ 1_i\rangle 1_{i+1}\rangle$	$ 1\rangle \otimes 0\rangle \otimes 0\rangle \otimes 1\rangle$

If we now allow the A species to be to couple to the C species, for a time $\tau_{CZ} = \frac{\pi}{8J}$. Up to a global phase, this yields the states listed below.

State of end spins	State of chain
$ 0_i\rangle 0_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$ 0_i\rangle 1_{i+1}\rangle$	$- 1\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$ 1_i\rangle 0_{i+1}\rangle$	$- 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 1\rangle$
$ 1_i\rangle 1_{i+1}\rangle$	$ 1\rangle \otimes 0\rangle \otimes 0\rangle \otimes 1\rangle$

The phases picked up are identical to those generated by the $\exp(-\frac{\pi}{4}\sigma_z^{(i)}\sigma_z^{(i+1)})$. This is the interaction part a CZ gate: By applying a local z-axis rotation through an angle of $\frac{\pi}{4}$, we recover the phases necessary for a controlled- Z gate.

State of end spins	State of chain
$ 0_i\rangle 0_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$ 0_i\rangle 1_{i+1}\rangle$	$ 1\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$ 1_i\rangle 0_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 1\rangle$
$ 1_i\rangle 1_{i+1}\rangle$	$- 1\rangle \otimes 0\rangle \otimes 0\rangle \otimes 1\rangle$

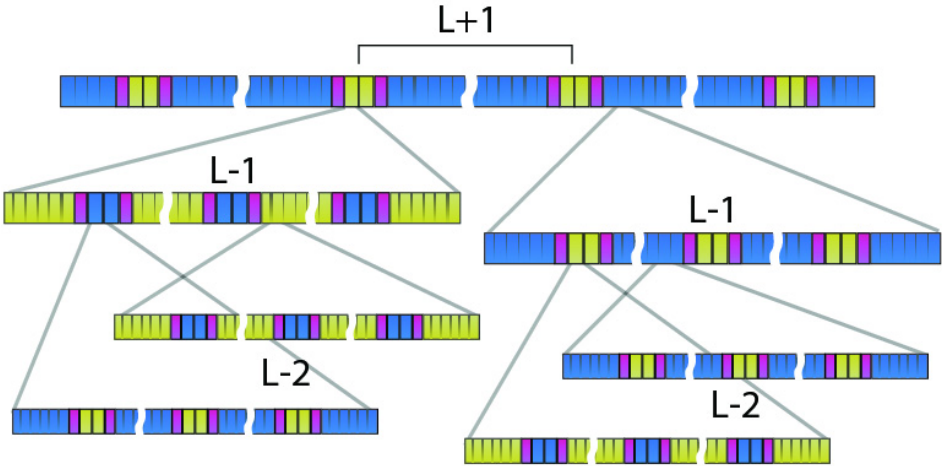


Fig. 5. Illustration of concatenation of levels $L + 1, \dots, L - 2$. All A (blue) and B (yellow) cells are concatenated with self-similar encoding patterns while C -reset cells (pink) are not encoded.

All that remains to complete the desired operation is to undo the mirror inversion of the B - C chain, and to undo the CNOT gate between the species A and C . This leaves the B - C chain in the zero state.

State of end spins	State of chain
$ 0_i\rangle 0_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$ 0_i\rangle 1_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$ 1_i\rangle 0_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$
$- 1_i\rangle 1_{i+1}\rangle$	$ 0\rangle \otimes 0\rangle \otimes 0\rangle \otimes 0\rangle$

As we can see the chain has only picked up a phase if both the end spins were in the $|1\rangle$ state, resulting in the desired CZ gate.

5 Higher concatenation levels

We have now seen how fault tolerant computing can be performed using one level of error correction. While this may well reduce the error rate, for arbitrarily long calculations to be possible it is necessary to concatenate the encoding. Many of the operations are identical between concatenation levels, but as mentioned earlier there are a number of differences. In this section we will discuss the necessary alterations to the chain layout and methods of replacing operations which do not generalise directly to higher levels of encoding. Figure 5 shows how the encoding used to protect the quantum states stored on the chain can be concatenated.

5.1 Chain layout

In order to concatenate the encoding, it is necessary to reproduce our chain layout at progressively higher levels of encoding. To do this, we will replace each species A qubit with an encoded qubit from the previous level. At the first level of concatenation, this will correspond to replacing each A -spin with a length of the structured chain, described in the last section, long enough to encode it and any associated ancillae. We will retain the mirror symmetric layout. It is necessary to also replace species B spins with encoded qubits to allow different levels of concatenation to be addressed separately. To accomplish this we will use the same encoding as for the A species meta spin-chain, but we will reverse the role of species A and B . Figure 5 illustrates this encoding scheme. The species C spins will remain unencoded. As these occur only once between encoded qubits, this will not cause trouble when proving the existence of a threshold for fault-tolerant computing.

5.2 Interacting species A and B

In order to perform gates at different levels of concatenation it will be necessary to perform controlled- Z gates between the qubits at the end of the the species A meta-chain and the nearest species B qubit. To do this, we will apply CZs one at a time between the qubits which make up the chain. In order to construct these gates, we will make use of the identity

$$(8) \quad CZ_{A,B} = CNOT_{A,C} CZ_{B,C} CNOT_{A,C} CZ_{B,C} \\ = H_C CZ_{A,C} H_C CZ_{B,C} H_C CZ_{A,C} H_C CZ_{B,C}$$

where the CNOT is controlled by the first index, and targetted on the second. Controlled- Z gates can be constructed from a $\sigma_z \sigma_z$ interaction, when combined with local rotations, and can be written as $\exp(i\frac{\pi}{4}(\sigma_z^{(B)} + \sigma_z^{(C)} - \sigma_z^{(B)} \sigma_z^{(C)}))$. Controlled- Z gates are their own inverse, so it follows that they can also be written as $\exp(-i\frac{\pi}{4}(\sigma_z^{(B)} + \sigma_z^{(C)} - \sigma_z^{(B)} \sigma_z^{(C)}))$. Substituting these result into the above equation yields

$$(9) \quad CZ_{A,B} = H_C e^{-i\frac{\pi}{4}(\sigma_z^{(A)} + \sigma_z^{(C)} - \sigma_z^{(A)} \sigma_z^{(C)})} H_C e^{-i\frac{\pi}{4}(\sigma_z^{(B)} + \sigma_z^{(C)} - \sigma_z^{(B)} \sigma_z^{(C)})} H_C \\ \times e^{i\frac{\pi}{4}(\sigma_z^{(A)} + \sigma_z^{(C)} - \sigma_z^{(A)} \sigma_z^{(C)})} H_C e^{i\frac{\pi}{4}(\sigma_z^{(B)} + \sigma_z^{(C)} - \sigma_z^{(B)} \sigma_z^{(C)})} \\ (10) \quad = H_C e^{i\frac{\pi}{4}(\sigma_z^{(A)} \sigma_z^{(C)})} R_z^{(C)} \left(-\frac{\pi}{4}\right) H_C R_z^{(C)} \left(-\frac{\pi}{4}\right) e^{i\frac{\pi}{4}(\sigma_z^{(B)} \sigma_z^{(C)})} H_C \\ \times R_z^{(C)} \left(+\frac{\pi}{4}\right) e^{-i\frac{\pi}{4}(\sigma_z^{(A)} \sigma_z^{(C)})} H_C R_z^{(C)} \left(+\frac{\pi}{4}\right) e^{-i\frac{\pi}{4}(\sigma_z^{(B)} \sigma_z^{(C)})}.$$

This form of the operator clearly shows that a CZ can be performed using only local rotations on C spins, together with the A - C and C - B Ising interactions. As the species C spins are isolated, it is trivial to perform local rotations upon them. Using this procedure for each pair of spins in the encoding, it is possible to perform transversal CZ gates between species A and B .

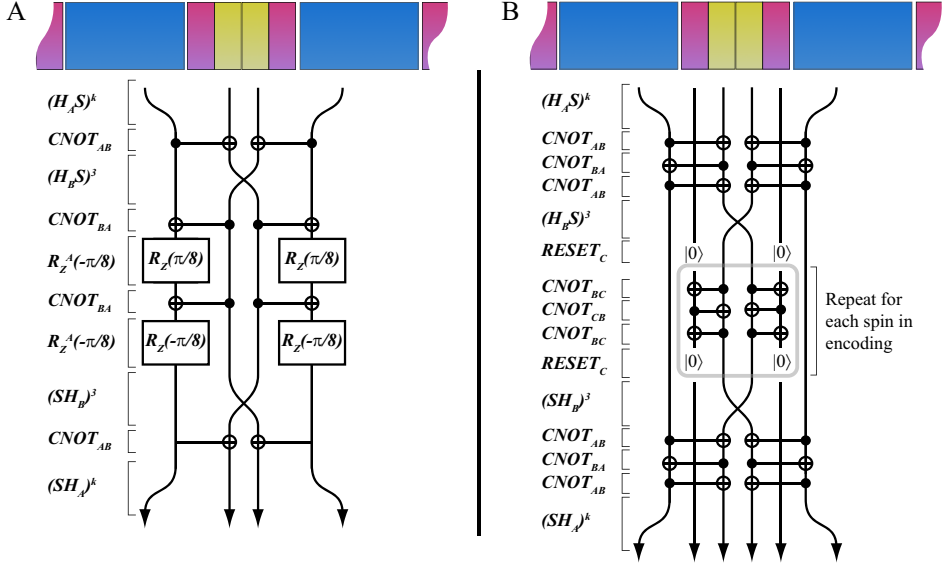


Fig. 6. Level dependent entangling and reset procedures. A) This shows the required gate sequence to delocalise the target qubits over the end spins of the species *A* subchain, and to interact adjacent *A*-subchain ends. B) This shows a level dependent reset procedure. The grey box indicates that, as only one spin can be reset at a time, the level-0 qubits within an encoding need to be swapped onto *C* and reset sequentially. The inline $|0\rangle$ indicate when the species *C* qubits are reset to the zero state.

5.3 Rotations on subchain end spins

Rotations on the end spins of the level-*L* meta-chain are required to be level dependent, and so will be accomplished using CZ gates as described in 5.2, conditioned on level-*L* species *B* meta-spins, together with global rotations. This procedure relies on the identity

$$(11) \quad R_z^{(ends)}(\theta) = CNOT_{B,A} X_B CNOT_{B,A} R_z^A \left(\frac{\theta}{2} \right) CNOT_{B,A} \\ \times X_B CNOT_{B,A} R_z^A \left(\frac{-\theta}{2} \right).$$

This identity holds, since $CNOT_{B,A} X_B CNOT_{B,A} X_B$ introduces an *X* operator on the first and last meta-spin in the species *a* subchain. The CNOT gates are trivially constructed out of the CZ gates previously described, combined with global Hadamard gates on species the *A* meta-spins. Thus the gating sequence required to produce a local rotation on the end of the chain is

$$(12) \quad R_z^{(ends)}(\theta) = H_A C Z_{A,B} H_A X_B R_z^A \left(\frac{\theta}{2} \right) H_A C Z_{A,B} X_B R_z^A \left(\frac{-\theta}{2} \right).$$

5.4 Controlled-Z gates

In order to achieve entangling gates, the CZ described in 5.2 will again be exploited. Figure 6(A) details the gating sequence required to perform a controlled-Z operation between the ends of the chain. This sequence results leads to an entangling gate equivalent to an Ising interaction applied for time $t = \frac{\pi}{4J}$. As we have already seen,

this is locally equivalent to a CZ gate.

5.5 Resetting ancillae

As we cannot introduce fresh ancillae, it is necessary to reuse the qubits within the chain. To achieve this, it is necessary to periodically reset the ancillary qubits. When this occurs, care must be taken not to destroy information at higher levels of encoding. As qubits will generally be delocalised at higher levels when it becomes necessary to reset qubits, we must have a reset procedure which depends on concatenation level.

A level dependant reset procedure is shown in figure 6(B). When resetting encoded qubits it is necessary to reset a single spin at a time. The spins must be swapped one at a time onto the species C spin, where they are reset.

6 Thresholds

We have now demonstrated how error correction can be performed within the chain, and how logic gates can be performed on encoded qubits without propagating errors within those encodings. This is, however, not enough to show that fault-tolerant computation is possible. To show that fault tolerant computation is in fact possible, it is necessary to show that there exists an error rate below which any errors can be suppressed arbitrarily by concatenation of the error correction code. This does not have to be the case, since concatenating codes may introduce errors at a higher rate than they are suppressed by the concatenation, in which case the error rate would balloon. As we will see, however, a threshold does exist below which errors can be suppressed by concatenation of the code.

6.1 Existence of a fault tolerant threshold

In order to prove the existence of a threshold for fault tolerant quantum computing within the system we will consider the error probability per gate at each level, L , of encoding P_L . For a code which can correct one error per encoded qubit $P_L = \kappa P_{L-1}^2$, for some constant κ which gives the number ways in which an uncorrectable error can arise, since all operations between qubits at level L use only level $L-1$ operations (which have error probability P_{L-1}), and at least two errors are required to produce an error which is not correctable at the present level of encoding. We will take N to be the number of level $L-1$ operations required to perform the level L fault tolerant operation requiring the most level $L-1$ operations, plus one round of level L error correction. Since the species C chain never increases in length the effective error rate per physical qubit when doing controlled- Z gates which cross the unencoded region is always constant, and bounded from above by 20ϵ (the number of physical operations required to swap a qubit onto and then off a C -spin), where ϵ is the error probability per physical qubit per operation. This means that N , as defined above, is independent of concatenation level, L . As κ is the number of ways in which an error uncorrectable at level $L-1$ can occur, it is strictly less than $\frac{N(N-1)}{2}$. This can

probably be made smaller, but it suffices to show a threshold. Thus $P_L < \kappa^{2^L-1} \epsilon^{2^L}$. As L goes to infinity, this limits to zero if epsilon is less than $\frac{1}{\kappa}$. Thus a threshold of $\frac{1}{\kappa}$ exists. While the encoding grows exponentially in size, the double exponential fall off in error rate leads negates the need for an exponentially long calculation[10].

References

- [1] D. Aharonov and M. Ben-Or. Fault-tolerant quantum computation with constant error rate. *arXiv quant-ph/9906129*, 1999.
- [2] S. C. Benjamin. Schemes for parallel quantum computation without local control of qubits. *Phys. Rev. A*, 6102:020301, 2000.
- [3] S. C. Benjamin. Simple pulses for universal quantum computation with a Heisenberg ABAB chain. *Phys. Rev. A*, 6405(5):054303, 2001.
- [4] S. C. Benjamin. Quantum computing without local control of qubit-qubit interactions. *Phys. Rev. Lett.*, 88(1):107904, 2002.
- [5] S. Bravyi and A. Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phys. Rev. A*, 71(2):22316, 2005.
- [6] A. Bririd, S. C. Benjamin, and A. Kay. Quantum error correction in globally controlled arrays. 2004. *quant-ph/0308113*.
- [7] J. L. Dodd, M. A. Nielsen, M. J. Bremner, et al. Universal quantum computation and simulation using any entangling hamiltonian and local unitaries. *Phys. Rev. A*, 65:040301, 2002.
- [8] Joseph Fitzsimons and Jason Twamley. Globally controlled quantum wires for perfect qubit transport, mirroring, and computing. *Phys. Rev. Lett.*, 97:090502, 2006.
- [9] A. G. Fowler, C. D. Hill, and L. C. L. Hollenberg. Quantum-error correction on linear-nearest-neighbor qubit arrays. *Phys. Rev. A*, 69:042314, 2004.
- [10] D. Gottesman. Fault-tolerant quantum computation with local gates. *J. Mod. Opt.*, 47:333–345, 2000.
- [11] Daniel Gottesman. The Heisenberg representation of quantum computers. 1998. *quant-ph/9807006*.
- [12] G. Ivanyos, S. Massar, and A. B. Nagy. Quantum computing on lattices using global two-qubit gates. *Phys. Rev. A*, 72:022339, 2005.
- [13] E. Jane, G. Vidal, W. Dur, et al. Simulation of quantum dynamics with quantum optical systems. *Quant. Info. & Comm.*, 3:15–37, 2003.
- [14] D. Janzing and P. Wocjan. Ergodic quantum computing. *Quant. Info. Proc.*, 4:129–158, 2005.
- [15] A. Kay. Error correcting the control unit in global control schemes. 2005. *quant-ph/0504197*.
- [16] A. Kay. Deriving a fault-tolerant threshold for a global control scheme. 2007. *quant-ph/0702239*.
- [17] J. Kempe. Approaches to quantum error correction. *Seminaire Poincare*, 2:1–29, 2005.
- [18] E. Knill, R. Laflamme, and W. Zurek. Accuracy threshold for quantum computation. (LAUR-96-2199), 1996. *quant-ph/9610011*.
- [19] J. Levy. Universal quantum computation with spin-1/2 pairs and heisenberg exchange. *Phys. Rev. Lett.*, 89:147902, 2002.
- [20] S. Lloyd. A potentially realizable quantum computer. *Science*, 261:1569, 1993.
- [21] R. Raussendorf. Quantum cellular automaton for universal quantum computation. *Phys. Rev. A*, 72:022301, 2005.
- [22] R. Raussendorf, J. Harrington, and K. Goyal. A fault-tolerant one-way quantum computer. 2005. *quant-ph/0510135*.
- [23] A. M. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, 77:793, 1996.

- [24] K. M. Svore, B. M. Terhal, and D. P. DiVincenzo. Local fault-tolerant quantum computation. *Phys. Rev. A*, 72:022317, 2005.
- [25] K.M. Svore, D. P. DiVincenzo, and B. M. Terhal. Noise threshold for a fault-tolerant two-dimensional lattice architecture. *Q. Inf. Comp.*, 7:297, 2007.
- [26] T. Szkopek, P. O. Boykin, H. Fan, et al. Threshold error penalty for fault-tolerant quantum computation with nearest neighbor communication. *IEEE Trans. on Nanotech.*, 5:42–49, 2006.
- [27] K. G. H. Vollbrecht and J. I. Cirac. Reversible universal quantum computation within translation-invariant systems. *Phys. Rev. A*, 73:012324, 2006.